Why Haven't More Quantum Algorithms Been Found?

PETER W. SHOR

AT&T Labs—Research, Florham Park, New Jersey

Abstract. I examine the question of why so few classes of quantum algorithms have been discovered. I give two possible explanations for this, and some thoughts about what lines of research might lead to the discovery of more quantum algorithms.

My discovery of the quantum factoring algorithm in 1994 caused great excitement among theoretical computer scientists. Quantum computers provided a completely new paradigm for the theory of computation, and this was the first time it had been shown that quantum computation could efficiently solve a problem that had already been established as important in this field. Many people expected a succession of other interesting algorithms to follow. The reality has been disappointing, especially compared with the progress of the rest of the field of quantum information processing. Experimental physicists have been proposing and exploring possible physical implementations of quantum computers at a pace far beyond what anybody but the most optimistic researchers originally expected. Quantum cryptography is coming of age, with several theoretical proofs of its security recently discovered, and with commercial quantum cryptography systems expected in the next year or so. The fields of quantum information theory and quantum complexity have been expanding dramatically, with a number of new interesting and important theoretical results. Meanwhile, the development of algorithms has lagged behind, with barely any significant new algorithms having been discovered in the last five years.

So far, all the quantum algorithms known to offer substantial speed-up over classical algorithms for the same problems fall into one of three classes. The first class uses the Fourier transform to find periodicity. This class contains the factoring and discrete logarithm algorithms [Shor 1997], Simon's algorithm [Simon 1997] (the first member of this class to be discovered), and Hallgren's algorithms for Pell's equation and certain other number theory problems [Hallgren 2002]. There is, in fact a different way of looking at the factoring algorithm that, although it yields basically the same algorithm, puts it into a setting that emphasizes spectral methods rather than periodicity [Kitaev 1996], but this approach has not yet yielded any new algorithms. The second class contains Grover's search algorithm, which can perform an exhaustive search of N items in $\sqrt{N}$ time [Grover 1997], and a number of extensions of this algorithm (see Grover and Sengupta [2002]). These extensions all have the general flavor of giving a square root improvement in the speed of optimization or search problems. The third class consists of algorithms for simulating or solving problems in quantum physics. This class contains Feynman's

Author's Address: P. W. Shor, AT&T Labs—Research, Room C237, 180 Park Ave., Florham Park, NJ 07932, e-mail: shor@research.att.com.

Journal of the ACM, Vol. 50, No. 1, January 2003, pp. 87–90.

original idea [Feynman 1982] of using quantum computers to speed up simulations of quantum physics. While not many theoretical papers have yet been written on this class of algorithms, it is clear that if quantum computers are ever developed, this class will be extremely useful in practice. Feynman came up with his idea of using quantum computers to simulate quantum physics in 1982, Simon's algorithm and the factoring algorithm were developed in 1993 and 1994, and Grover came up with his original search algorithm in 1995. Since then, there have been further theoretical developments within each of these classes of algorithms, but no new classes of quantum algorithms have been discovered.

As the discoverer of the quantum factoring algorithm, one of the questions I am often asked is why there are so few quantum algorithms known that offer speed-up over classical algorithms. The answer I usually give is that I don't know, but that I can think of two possible reasons that this might be the case. The first possible reason is that quantum computers operate in a manner so different from classical computers that our techniques for designing algorithms and our intuitions for understanding the process of computation no longer work. The second reason is that there really might be relatively few problems for which quantum computers can offer a substantial speed-up over classical computers, and we may have already discovered many or all of the important techniques for constructing quantum algorithms. This article contains an expansion of these thoughts.

Both of these explanations address the question of why we haven't seen more speed-ups from quantum algorithms, and I believe both of these explanations are likely to be true to a greater or lesser extent. It is certainly true that quantum computers are very difficult to reason about using classical intuition. Physicists have spent decades developing their intuitions about quantum phenomena, and many of the techniques they use came decades after the original development of quantum mechanics. Computer scientists, on the other hand, have been thinking about quantum mechanics for barely a decade. Any quantum algorithm offering a speed-up over classical computation must use interference; this phenomenon is unknown in classical computer science, and most theoretical computer scientists are not used to reasoning about it. Thus, it seems quite likely that several new and significant quantum algorithmic techniques have yet to be discovered.

On the other hand, much of the research into new quantum computer algorithms has been spent looking for superpolynomial speed-ups. While these do not occur in Grover's algorithm and its extensions, they can occur in the classes of quantum algorithms that use periodicity finding and that simulate quantum physics. Superpolynomial speed-ups cannot arise from problems that have polynomial-time classical algorithms, so researchers have been concentrating on problems that are not in the classical computational class P. The first class of problems not in P that come to mind are the NP-complete ones. A quantum algorithm solving NP-complete problems in polynomial time would be a momentous discovery, but I believe that the most likely scenario is that this is not possible. It has been proved that there is an oracle, relative to which NP-complete problems cannot be solved in polynomial time [Bennett et al. 1997], and while there are fewer reasons supporting the belief that quantum computers cannot solve NP-complete problems than there are supporting the nearly universal belief that classical computers cannot solve them, many researchers are still pessimistic that quantum computers can solve NP-complete problems.

If we assume that NP-complete problems are not solvable efficiently on a quantum computer, then in order to achieve a superpolynomial speed-up, we must look within the class of problems which are neither NP-hard nor in P. There are only a relatively small number of well-studied problems that are suspected to be in this class. No general theory is known for these problems, and relatively few of them are known to be reducible to each other, so they all must be considered individually. It may be that many of these problems do not indeed have polynomial-time algorithms on quantum computers. People have to date concentrated their efforts on those problems that appear to have structure related to periodicity, thus providing a possible means of attack. These include the problems of graph isomorphism and that of approximating short vectors in a lattice. Neither of these problems has yet yielded to a quantum attack.

Part of the expectations for the discovery of many quantum speed-ups may be due to analogies with the history of classical computation. After the identification of NP-completeness [Cook 1971, Karp 1972, Levin 1973], there followed a plethora of papers classifying problems either as polynomial time (giving efficient algorithms) or as NP-hard. This may have raised our expectations too high, as the success of this classification effort has now left relatively few well-studied problems that are not known to be in one of these two classes. By searching for superpolynomial speed-ups, we may also be attempting too difficult a task. By contrast with researchers in quantum computing, researchers in classical algorithms spend their time not only trying to put more problems into the class P, but also trying to discover faster algorithms for problems that are already known to be in P. By trying to discover new ways of solving problems already known to be in P, researchers have often been able to find new and fruitful techniques, which then can be used to help solve other problems, sometimes including ways of efficiently solving problems not known to be in P.

One research area that might be worth exploring is to try to find faster quantum algorithms for problems already known to be classically solvable in polynomial time. This approach is limited to providing polynomial factor speed-ups. While this is certainly less exciting than finding superpolynomial speed-ups, and is also less likely to yield practical results—since quantum computers are likely to be slower than classical computers—it could nevertheless yield new techniques for designing quantum algorithms. At this point, my belief is that any new techniques have the potential to be of great value in further exploration of quantum algorithms, and, if this avenue can help discover such new techniques, it should be pursued.

REFERENCES

BENNETT, C. H., BERNSTEIN, E., BRASSARD, G., AND VAZIRANI, U. V. 1997. Strengths and weaknesses of quantum computing. *SIAM J. Comput. 26*, 1510–1523.

COOK, S. 1971. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*. ACM, New York, 151–158.

FEYNMAN, R. 1982. Simulating physics with computers. *Internat. J. Theoret. Phys. 21*, 467–488.

GROVER, L. K. 1997. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett. 78*, 325–328.

GROVER, L. K., AND SENGUPTA, A. M. 2002. From coupled pendulums to quantum search. In *Mathematics of Quantum Computation*, R. K. Brylinski and G. Chen, Eds. Chapman & Hall/CRC, Boca Raton, FL, 119–134.

HALLGREN, S. 2002. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. ACM, New York, 653–658.

KARP, R. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum, New York, 85–103.

KITAEV, A. YU. 1996. Quantum measurements and the Abelian stabilizer problem. ECCC Report TR96-003. Los Alamos archive, e-print quant-ph/9511026.

LEVIN, L. 1973. Universal search problems (in Russian). *Prob. Pered. Inf. 9*, 3, 265–266.

SHOR, P. W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput. 26*, 1484–1509.

SIMON, D. R. 1997. On the power of quantum computation. *SIAM J. Comput. 26*, 1474–1483.